Int. J. Nonlinear Anal. Appl. 14 (2023) 1, 2279–2300
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.27197.3529

Human-Whale cooperation optimization (HWO) algorithm: A metaheuristic algorithm for solve optimization problems

Farnoosh Parandeh Motlagh[a], Vahid Khatibi Bardsiri[b,*], Amid Khatibi Bardsiri[b]

[a]*Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran*
[b]*Department of Computer Engineering, Bardsir Branch, Islamic Azad University, Bardsir, Iran*

(Communicated by Mohammad Bagher Ghaemi)

Abstract

Metaheuristic algorithms are one of the most effective methods for solving optimization problems and are modeled on the behavior of living things or biological phenomena. The swarm behavior of animals in nature to survive is a good way to create metaheuristic algorithms with a group intelligence approach. The swarm hunting mechanism is one of the most interesting meta-behavioral behaviors observed in a large number of organisms, and the chances of success in prey hunting by swarm behaviors will increase. In this paper, a new metaheuristic algorithm with a swarm intelligence approach is presented by using the human hunting mechanism and whale. In this type of behavior, whales and humans participate in hunting in such a way that whales and humans benefit from each other. Implementation and analysis of the proposed method provided less error than 82.60% of the experiments of other algorithms such as particle swarm optimization(PSO), firefly algorithm(FA), grasshopper optimization algorithm(GOA) and butterfly optimization algorithm(BOA). Experiments show that the proposed method converges in complex functions with a probability of 4.36% in local optimizations, which is less than the comparable algorithms. Experiments show that the proposed method can be implemented on a wide range of functional optimization problems and reduces the optimization error due to the simultaneous local and global search of the intelligent algorithm.

Keywords: optimization problems, benchmark function, metaheuristic algorithms, swarm intelligence algorithms, human-whale cooperation optimization
2020 MSC: 68T99

1 Introduction

Metaheuristic algorithms are a set of methods for solving optimization problems that are modeled on the modeling of phenomena in nature and the problem-solving method of animals or physical phenomena [13]. Modeling in metaheuristic algorithms involves using the survival mechanism or finding the optimal answer in nature that is used by animals or the laws of nature. Most of the problems in nature that organisms face are optimization problems because in these problems each organism tries to find the most optimal solution for finding food, hunting, and making food [4, 18]. Most of the behavior of animals in nature is swarm behavior because they have learned over time that living in a group increases their chances of survival. Studies show that in order to find food, they first consider the routes to

*Corresponding author
Email addresses: farnoosh.pm16@gmail.com (Farnoosh Parandeh Motlagh), kvahid2@live.utm.my (Vahid Khatibi Bardsiri), khatibi_amid@yahoo.com (Amid Khatibi Bardsiri)

Received: November 2021 *Accepted:* February 2022

be random and at the same time release an acidic substance called pheromone in their path [2, 29]. Their release of pheromones or acid causes each ant to know the return paths, and on the other hand, the ants move in paths where more pheromones have been released. The tracking behavior of the released pheromones causes them to choose short paths to food and nests. Ant behavior in finding food is an example of swarm intelligence behavior and inspires the ant optimization algorithm. The ant optimization algorithm can be used in graph-based optimization problems such as Travelling Salesman Problem (TSP) [7], which is an NP-Hard problem that can be solved by this algorithm. The swarm hunting mechanism in gray wolves and the attack on one side of the prey is an example of a metaheuristic algorithm with a swarm hunting approach among animals that inspires the gray wolf optimization(GWO) algorithm [24]. swarm hunting behavior of Humpback whales in the whale optimization algorithm(WOA) [26], swarm behavior of spotted hyenas in the spotted hyena optimizer(SHO) algorithm [11], swarm behavior of hawks in group hunting and prey siege in harris hawks optimization (HHO) algorithm [15]. Swarm hunting behavior can also be seen in insects and arthropods and is not unique to mammals. Studies show that insect behavior in swarm hunting is highly intelligent, and therefore a large number of metaheuristic algorithms of this type are considered, including the social spider optimization(SSO) algorithm [8], antlion optimizer (ALO) algorithm [16], black widow optimization(BWO) algorithm [14], dragonfly algorithm (DA) [31] was mentioned. Metaheuristic algorithms are divided into categories based on their nature and type according to Figures 1, such as evolutionary algorithms, swarm-based algorithms, physical-based algorithms, and human-based algorithms [25]. Evolutionary algorithms use genetic and evolutionary rules to solve problems and find the optimal answer, a clear example is a genetic algorithm(GA). Swarm intelligence algorithms can emulate a variety of social inspire, and these inspired are not necessarily hunting-based and can be a variety of factors, including communicating with light in firefly algorithms(FA) [20], sound communication in the bat algorithm(BA) [22], communication and swarm heating in the empirical penguin optimization(EPO) algorithm [9]. In some metaheuristic algorithms, each member of the population can use individual and swarm memory to find the optimal solution, an example of which can be seen in the particle swarm optimization(PSO) algorithm [3]. In some cases, group intelligence algorithms include insect behavior to find food as a group, such as the fruit fly optimization algorithm [6]. Studies show that the behavior of some species is very intelligent and clever, and in this type of behavior, an organism acts in such a way that its behavior alone is highly intelligent and interacts less with existing species. In this type of behavior, intelligence does not mean group participation, but rather a type of intelligent behavior on its own. For example, octopus behavior is highly intelligent in finding food, and the octopus optimization algorithm(COA) [28] is one of these behaviors in nature. As another example in nature, some fish species have created beautiful lines and shapes on the seabed to guide other fish to mate, and the Circular structures of pufferfish (CSOPF) algorithms [5] have been coded accordingly. In metaheuristic methods with a physical approach, in most cases, physical laws are modeled and efforts are made to extract optimal solutions based on these rules. Examples of metaheuristic algorithms with a physical approach include water evaporation optimization [19], electromagnetic field optimization [1], lightning search algorithm (LSA) [17] and Gravitational search algorithm(GSA) [21]. Human behavior-based algorithms also use human interactions and relationships to solve the optimization problem, and in most cases, the political, cultural, and educational approach is discussed. These algorithms are more intelligent, but their modeling complexity is more complex. Examples of these algorithms are the poor and rich optimization(PRO) algorithm [27], teaching–learning-based optimization (TLBO) [33], and football game algorithm [12]. of course, this classification is not complete for the meta-heuristic algorithm because it does not classify behaviors such as plant behavior, while a number of metaheuristic algorithms are modeled on plant behavior, such as sunflower optimization (SFO) algorithm [32] and Forest optimization algorithm [30]. In some cases, transcendental algorithms are modeled on the behavior of monocellular organisms and can be grouped into biological or group categories, such as bacterial algorithm [35]. In most studies, metaheuristic algorithms are classified as swarm intelligence, evolutionary, physical, and biological algorithms.

Evolutionary behaviors are overly random, and due to mutation and crossover operations performed on qualified members and convergence to optimal local in the first iteration. The challenge with physical algorithms is that they are not population-based and they have complex relationships, but in practice, these relationships complicate modeling and lack intelligence, and like swarm intelligence algorithms, population members do not interact and exchange knowledge. swarm intelligence algorithms are an important category of meta-heuristic algorithms, and in these algorithms, a social and group nature is used in organisms to solve problems.

An important challenge for these algorithms is that they rely too much on the right member or the most appropriate solution to the problem, and in most of them, like the whale optimization algorithm, the members of the population circle and search around the best solution, while it is necessary that search the problem area more and better and of course smart, however, the problem space needs to be searched more intelligently. Human behavior-based algorithms use human behavior to solve problems to find the optimal solution. The main weakness of these methods, such as swarm intelligence algorithms, is that the members of the population rely too heavily on the best solution. In these

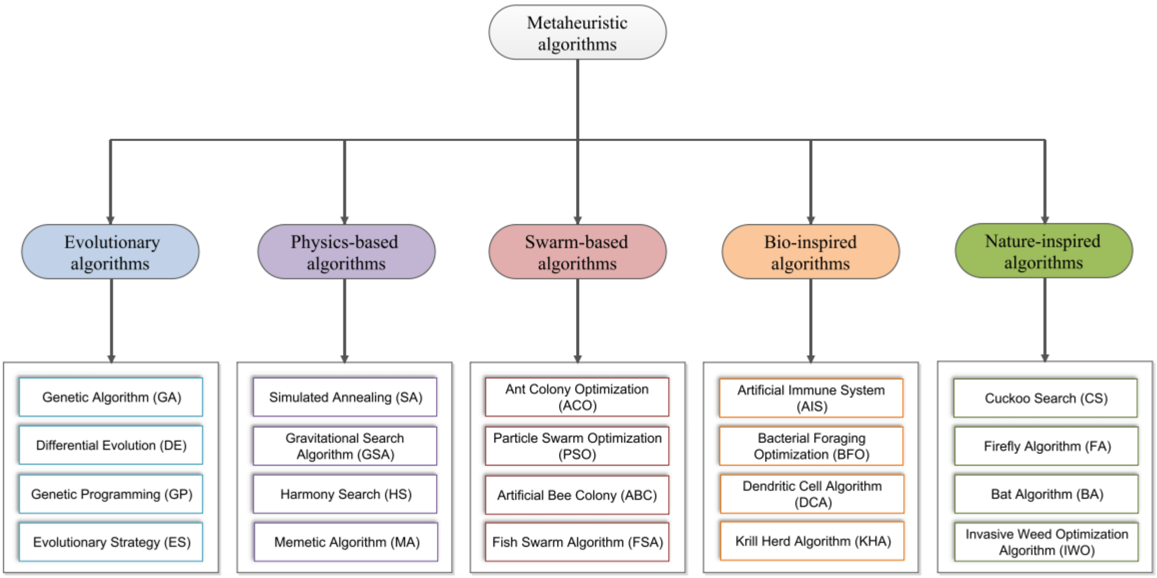

Figure 1: Categorization of a variety of metaheuristic methods based on problem solving approach [10]

algorithms, there is a possibility of deceiving members and converging to local optimal. In this paper, a metaheuristic algorithm with a swarm intelligence approach based on human and animal behavior is presented, and this algorithm is one of the limited methods of swarm intelligence that simultaneously uses the behavior of two different species to solve optimization problems. In this article, the intelligent behavior of humans in fishing with the help of whales near the coast of Australia, which is a common practice among the natives of Australia, is modeled and formulated to create a meta-functional method. This article has been prepared and compiled in several sections. First, the proposed algorithm is formulated and modeled. Then, the proposed algorithm is implemented on a number of standard benchmark functions and then its accuracy is compared with several metaheuristic algorithms. In the final part of the article, the proposed algorithm is implemented and evaluated on several practical optimization problems.

2 Human-Whale Cooperation Optimization Algorithm

There are groups of whales that have complex social behavior and a high level of intelligence for interaction and cooperation between themselves and other species. The whales near the coast of Australia are found in oceans around the world, from the Arctic to the tropics. The whales choose different prey as food, however, some groups only look for specific prey such as fish. Some feed only on fish, while others attack flocks of marine mammals, such as jaws, sea lions, guinea pigs, and even whales. They are at the top of any food pyramid, generally have no natural enemies, and sometimes prey on large sharks. Figure 2 and Figure 1 show a view of these large creatures that live in the oceans. They are at the top of the food pyramid, generally have no natural enemies, and sometimes prey on large sharks. Figure 2 shows the different behaviors of these giant creatures. The whales are very social, and some of them have mother-based families that make up the most stable of all animals. There is a kind of collective collaboration between humans and these whales in the behavior of a number of whales near the coast of Australia that live on the shores of Australia. Here hunters are placed in different parts of the coast with the help of small boats and are waiting for the herds of fish that are being led by the whales to the fishing boats [34].

Figure 2 shows the swarm behavior of whales hunting and collaborating with Indigenous Australians:

Over time, whales have learned to attack groups of fish in swarms and steer them to the fishermen's boats so that they can catch the fish. Fishermen use the rest of the fish to reward whales. The mechanism of human hunting with the help of giant creatures such as whales is interesting and very effective in its kind and makes fishermen achieve a large volume of hunting. Fishermen's and whales' methods of catching fish are efficient and effective and can inspire NP-hard problems to solve. Therefore, in this research, the collective and swarm behavior of humans and whales in fishing is used to create swarm intelligence. In the proposed method, each solution to the problem is coded in the form of whales or humans and attempts are made to bring the population closer to the optimal point, which is the center of fish accumulation. The advantage of modeling the proposed method can be summarized as follows:

Figure 2: Cooperation between humans and whales in fish hunting

- So far, there has been no swarm intelligence system based on the behavior of humans and organisms to create a mass intelligence algorithm.
- The whale swarm intelligence algorithm, which is a collaboration between humans and whales, has not yet been investigated.
- The proposed algorithm has intelligent behaviors on the part of humans and whales, and therefore the problem space is well searched.

The proposed algorithm is fundamentally different from the whale optimization algorithm because there is no discussion of human and whale cooperation in the whale optimization algorithm, on the other hand, in the whale optimization algorithm, the problem-solving mechanism is the production of bubbles and rotational and spiral motions. Finally, the whale optimization algorithm is for humpback whales, but the proposed method is considered for the whale, which is a different species. In this section, the behavior of humans and whales in hunting is modeled to provide a proposed and new swarm intelligence algorithm.

2.1 Coding solutions

In the first phase of this study, the behavior of swarm intelligence hunting between humans and whales for fish and dolphin hunting is reviewed and an attempt is made to provide a metaheuristic algorithm with a global and local search approach to be able to extract optimal solutions with high accuracy. For this purpose, humans and whales are considered solutions to the problem of optimization. In this algorithm, based on the cooperation of whales and humans, an attempt is made to find the optimal space for the problem. In this mechanism, the optimization problem solutions are encoded as a whale or human, and each of them explores the problem space. In the proposed method, it is assumed that there is a population of humans and whales, and each of them is considered a solution to the problem. There is no difference between a whale and a human being in practice, and each of them is considered a solution to the problem, but whale-type solutions are those solutions that are more fitness because these solutions are close to the optimal points or swarm of fish. In the proposed method, the population members are expressed in the form of a set of answers according to Eq. (2.1), some of which are whales and some of which are human:

$$P = \{W_1, W_2,\ \ldots,\ W_k, \ldots, H_1, H_2,\ \ldots,\ H_m\} \tag{2.1}$$

In this equation, the members of the whale and human population are placed in a set, and their number is assumed to be k and m, respectively, and $n = k + m$, where n is the number of members of the proposed algorithm population. In the proposed method, it can be assumed that initially some of the solutions are whales and some are humans. Here, for simplicity, half of the population can be considered first whales and half of them as humans, and this has been formulated in Eq. (2.2):

$$k = m = \frac{n}{2} \tag{2.2}$$

In the proposed method, the concept of merit is used to select whales and humans. In this study, members of the more deserving population can be considered whales and more unworthy members can be considered human beings. With this mechanism, half of the population in each iteration is considered to be a whale and the other half is human.

In this case, the problem space is divided into two categories, including whales and humans, and whale members are more worthy because they are closer to the fish category. On the other hand, humans are far from fish and try to get closer to these centers. n modeling the proposed method, considering that the optimal point is the gathering place of the fish group, it is assumed that the human population is directed to this point and here we can assume with a simple assumption that the position of the most optimal whale is the estimated position of the swarm gathering. In the proposed method, it is assumed that humans or non-optimal solutions move towards it. In the proposed method, according to the Eq. (2.3) and Eq. (2.4), the population of whales and humans can be placed in two primary population matrices, and then each member of the population can be evaluated by the evaluation function which is shown here by f function. Using the evaluation function, each member of the whale and human population is evaluated according to the Eq. (2.5) and Eq. (2.6):

$$P_W = \begin{bmatrix} W_{1,1} & W_{1,2} & \dots & W_{1,d} \\ W_{2,1} & W_{2,2} & \dots & W_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ W_{k,1} & W_{k,2} & \dots & W_{k,d} \end{bmatrix} \tag{2.3}$$

$$S_H = \begin{bmatrix} H_{1,1} & H_{1,2} & \dots & H_{1,d} \\ H_{2,1} & H_{2,2} & \dots & H_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ H_{m,1} & H_{m,2} & \dots & H_{m,d} \end{bmatrix} \tag{2.4}$$

$$f_W = \begin{bmatrix} f([W_{1,1}, W_{1,2}, \ \dots, W_{1,d}]) \\ f([W_{2,1}, W_{2,2}, \ \dots, W_{2,d}]) \\ \vdots \\ f([W_{k,1}, W_{k,2}, \ \dots, W_{k,d}]) \end{bmatrix} \tag{2.5}$$

$$f_H = \begin{bmatrix} f([H_{1,1}, H_{1,2}, \ \dots, H_{1,d}]) \\ f([H_{2,1}, H_{2,2}, \ \dots, H_{2,d}]) \\ \vdots \\ f([H_{m,1}, H_{m,2}, \ \dots, H_{m,d}]) \end{bmatrix} \tag{2.6}$$

In these relations, d is the number of dimensions of the problem and the function of the target, and population of whales and humans are considered with the P_W and S_H matrices, respectively. In these relationships, the fitness of each whale and human population is considered with f_W and f_H matrices, respectively. Each of the population matrices is updated regularly in each iteration of the proposed method to search the space under the objective function.

2.2 Estimating position of fish or prey

In the proposed method, it is necessary to provide a mechanism that the position of the fish handle or the optimal point can be estimated or calculated approximately so that it can be considered as a gathering place for the fish handle. In the proposed algorithm, the position of the worthy members of the population or the whale is used and their average position is considered as the optimal point. Assume that the whale population is considered to be a set of $W = \{W_1, W_2, \ \dots, \ W_k\}$. The average or average position of whales is estimated to be the accumulation of fish, and the Eq. (2.7) can be used:

$$\overline{W} = \frac{W_1 + W_2 + \ \dots + \ W_k}{k} = \frac{\sum_{i=1}^{k} \ W_i}{k} \tag{2.7}$$

Each member of the set has the position W the position of a whale in the problem area and displays the location and position of the fish. Each population of the human group, to hunt the fish group, can consider the center of the fish group and move towards it to reach the optimal point and search for it. Each member of the human population does not necessarily choose the optimal point to go to the optimal point but defines a vector, and according to this, the random vector moves toward the optimal point. This mechanism in the proposed method avoids excessive local search for the current optimal solution and reduces local optimal. Another way to select the position of the swarm of fish is to use the position of the optimal population whale, which can be displayed here with. Here the most optimal whale is an exploration to estimate the optimal position or location of the fish that is the optimal solution here.

2.3 Directing the hunter to the fish group

In the proposed method, every human solution tries to move towards the whales, but this movement is not necessarily in the direction of a single whale and can be in the direction of the worthy whales of the population. Here it is assumed that in the first iteration there are k whales and each human solution moves in the direction of k whales in terms of iterations. Here, in terms of repetition, the number of this k position is reduced so that the nature of the search in the last iterations is more based on more optimal solutions, and the search changes from global to local search over time. In Figure 3, a human-type solution for hunting fish calculates its distance from the whales of the population and then moves in the direction of the outcome of these vectors to the fish groups that can be around the whales. In the proposed method and the last iteration, only two whale positions are used, and these whales are considered the most worthy whales in the population:

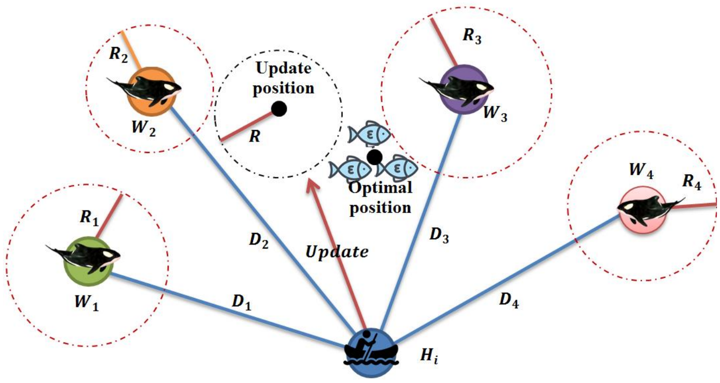

Figure 3: Directing a hunter to the whales

It has been observed that a non-optimal solution such as H_i to move towards whales and chase fish or the same optimal solution initially selects a number of optimal wall populations. To select these whales, the roulette wheel mechanism can be used, and then any human or non-optimal solution can calculate its distance from a whale such as W_j as a Eq. (2.8):

$$\overline{W} = \frac{W_1 + W_2 + \cdots + W_k}{k} = \frac{\sum_{i=1}^{k} W_i}{k} \tag{2.8}$$

In order to increase the dynamism and intelligence of the search algorithm, we can define a search radius around whale W_j and the size of R_j and calculate the distance H_i from this circle to the center of the W_j whale and as a Eq. (2.9):

$$D_{ij} = \left\| R_j W_j - H_i \right\| \tag{2.9}$$

In this Eq., the calculation of the human distance from the whale is affected by the R_j factor, which can be a random number between zero and one. On the other hand, this can be suggested as a Eq. (2.10):

$$D_{ij} = \|W_j \pm R_j - H_i\| \tag{2.10}$$

To update the position of a human-type solution or a hunter such as H_i in the direction of the whale such as W_j, the Eq. (2.11) can be used:

$$H_i = W_j + \theta \cdot D_{ij} \tag{2.11}$$

If we expand this relation according to Eq. (2.11), then to change the displacement of the predator of Eq. (2.12), the following is presented:

$$X_i = W_j + \theta \cdot \|W_j \pm R_j - H_i\| \tag{2.12}$$

In this Eq., θ plays an important role in changing the nature of the search from global to local. With this mechanism, in the last iterations, most of the space around the optimal solutions can be searched by local search, and in the first iteration, most of the search is considered as a global search so that the algorithm is less involved in local search. Nonlinear relationships can be used to model the θ parameter. Here, using the cosine function, this parameter can be dynamic and not changed by iteration. To change the θ parameter in the proposed algorithm, we can use Eq. (2.13):

$$\theta = \theta_0 - a \cdot \cos f_0\left(\frac{\pi}{2} \times \frac{It}{\text{MaxIt}} - \frac{\pi}{2}\right) \tag{2.13}$$

In this regard, the parameter θ_0 and its initial value are considered to be 3, and It and MaxIt are the current iteration number and the maximum iteration of the proposed algorithm, and a is a coefficient to reduce the graph. Here, a hunter-type solution moves in the direction of a whale, and this displacement is equal to the value of X_i, and if the number of whales to be moved in their direction is equal to L. The displacement equation can be proposed as a Eq. (2.14), and this equation can be expanded more accurately as Eq. (2.15):

$$H_i = \frac{\sum_{i=1}^{L} X_i}{L} \tag{2.14}$$

$$H_i = \frac{\sum_{i=1}^{L} W_j + \theta \cdot \|W_j \pm R_j - H_i\|}{L} \tag{2.15}$$

L is the number of whales used by a hunter for tracking in each iteration, and their number can be reduced in each iteration so that in the final iterations, only the optimal whales can be searched.
Here it is assumed that in the first iteration of the algorithm, the total population of whales in the number of k is used, and in the last iteration of the algorithm, only their $k/4$ is used. The number of L in each iteration can be suggested by the Eq. (2.16):

$$L = k - \frac{3}{4}\Big(\frac{it-1}{\text{MaxIt}-1}\Big)k \tag{2.16}$$

By gradually reducing the L parameter, the hunter can look for more optimal solutions in the final iteration.

2.4 Besieging fish

In addition to guiding predators, whales try to lead them to a specific area by rotating them around a fish handle to make it easier for them to hunt. In the proposed method, it is assumed that the whale gathering center or the most optimal position of the whales is considered as the gathering of fish, and each of the whales made circular movements around it so that they could surround them. This behavior is an attempt to locally search for the optimal whale or optimal solution and can be simulated by the triangular relations of sine or cosine as shown in Figure 4:

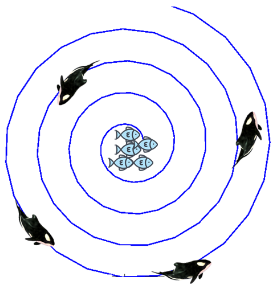

Figure 4: Siege of fish by whales

Triangular laws such as sine are used to model the behavior of the siege. An example of this equation can be given in Eq. (2.17):

$$W_i = D^* \exp(bp) * \sin(2\pi p) \ + W^* \tag{2.17}$$

In this regard, W^* is the optimal whale position, b and p are e random number in the range $[-1, +1]$ and b is a random number between zero and one, and D is the distance between the W_i whale and the optimal whale. As a Eq. (2.18), it is defined:

$$D = |W^* - W_i| \tag{2.18}$$

To be more effective, the siege is carried out with two different strategies, such as Eq. (2.19), and 50% of the siege cases are considered to be the most optimal whale, and in 50% of the siege cases, the average population is considered:

$$W_i = \begin{cases} |W^* - W_i| \,.e^{bp}\sin(2p\pi) \ + W^* & r > 0.5 \\ \left|\overline{W} - W_i\right| .e^{bp}\sin(2p\pi) \ + \overline{W} & r \leq 0.5 \end{cases} \tag{2.19}$$

r is a random number between zero and one that causes the algorithm to switch between two types of searches with equal probability. This strategy makes the search behavior of the proposed algorithm not necessarily local search and is searched simultaneously and with equal probability about the optimal point and the point of gravity of the population.

2.5 Hunter attack

In the proposed method, and especially in the final iterations, it is necessary for the predatory population to attack the swarm of fish that have been caught. Initially, a small number of hunters attack the fish population to perform a nationwide search mechanism, and then, in the final iteration, the number of hunters is steadily increasing to further search for worthy members or worthy whales. In the proposed method, the number of hunters who attack the fish swarm or optimal solutions in each iteration is calculated from Eq. (2.20), and this equation can be shown more simply and in the form of Eq. (2.21):

$$A = m - L \tag{2.20}$$

$$A = m - k + \frac{3}{4}\left(\frac{it - 1}{\text{MaxIt} - 1}\right)k \tag{2.21}$$

In these relationships, m is equal to the number of hunter populations, k is the number of whales, and A is the number of hunters who attempt to attack prey or optimal areas such as the Eq. (2.22) in each iteration :

$$H_i = W^* + \left(1 - \frac{it}{\text{MaxIt}}\right)(W^* + R) \tag{2.22}$$

R The search radius is considered to be the optimal solution and can be continuously reduced to increase the effectiveness so that the nature of the search gradually changes from one place to another according to iteration. In the proposed method, in order to be able to identify local or global search, the S factor such as Eq. (2.23) can be used:

$$S = \theta \times \text{rand}\,(-1, +1) * \left(1 - \frac{it}{\text{MaxIt}}\right) \tag{2.23}$$

The value of S varies frequently between the value of the interval $[-\theta_0, +\theta_0]$ an example of which can be seen in the random behavior of this function, as shown in Figure 5. According to the diagram, it can be seen that its value is constantly decreasing, and if the value of $|S|$ is more or less than one, the search will be global and local, respectively.

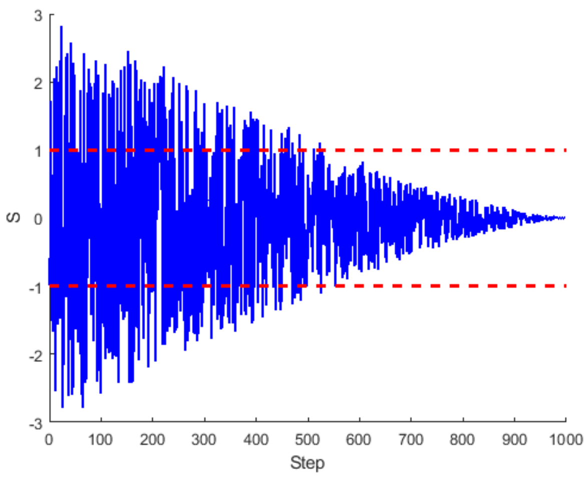

Figure 5: Factor search in HWO

2.6 Flowcharts and pseudo code

Figure 6 and 7 show the pseudo code and flowchart, respectively, the proposed method or whale optimization algorithm. The proposed method to find the optimal solution first initializes the initial parameters such as the initial population and the number of repetitions. In the second stage, an initial population of random solutions is created and half of the solutions that are more worthy are considered whales, and other solutions are considered hunters. In the next step, the position of the fish mass is assumed to be based on the position of the average whale or optimal whale point. In the next step, the hunters move toward the prey, then the siege is surrounded by whales with rotational movements, then the whale population and the hunter are regularly updated, and finally, the best member is transferred to the output as the optimal solution.

Pseudo code of Human-Whale Cooperation Optimization (HWO) Algorithm

```
Set: The population size,Max Iterationt, Dim, it=1 , BestSol.Cost=inf, etc
Initialize the random population X_i(i = 1,2,...,N) & Calculate the fitness values of population
Population is divided into whale an hunter W_i(i = 1,2,...,N/2) & H_i(i = 1,2,...,N/2)
while(It<=MaxIt)
    L = k - 3/4 ((it-1)/(MaxIt-1))k  % Calculate the number of Helping Whales
    θ = θ_0 - a.cos(π/2 × It/MaxIt - π/2) %Calculate Hunter Search Factor
    S = θ × rand(-1,+1) * (1 - it/MaxIt) % Calculate global or local search coefficient
    Sum=0
    for i=1: N/2
        Sum = Sum + W_i % Calculate the average position of whales
    end for
    W̄ = Sum/(N/2)
   for i=1:N/2 % Directing hunters to whales
    for j=1: L
        R_j = rand
        if rand>0.5
            D_ij = ||R_j W_j - H_i|| & X_i = W_j + θ.D_ij
        else
            D_ij = ||W_j ± R_j - H_i|| & X_i = W_j + θ.D_ij
        endif
    end for
      H_i = (Σ_{i=1}^{L} X_i)/L or H_i = (Σ_{i=1}^{L} W_j+θ.||W_j±R_j-H_i||)/L % Hunter moves along several whales
if Cost(H_i) better than BestSol.Cost
  BestSol=H_i or W*=H_i
end for
for i=1:N/2 %Collaboration between hunters
    A=randperm(N/2), A(A==i)=[], a=A(1), b=A(2), c=A(3)
    H_i=H_a+rand(H_b - H_c)
end for
    for i=1: N/2
         if rand>0.5
            W_i=W_i+rand.(W* - W̄) %Search for optimal point by whales
        else
            W_i=W_i+rand.(W* - W_i) %Search for gravity point by whales
        endif
    end for
    for i=1: N/2
         if |S|<1
           if rand>0.5
             W_i = |W* - W_i)|.e^{bp} sin(2pπ) + W* %Rotate search around the optimal point by whales
           else
             W_i = |W̄ - W_i)|.e^{bp} sin(2pπ) + W̄ %Rotate search around the optimal point by whales
           endif
        else
            Random search
        endif
Update BestSol or W* & It=It+1
end for
end while
Output W* or BestSol
```

Figure 6: Pseudo code of Human-Whale Cooperation Optimization (HWO) Algorithm

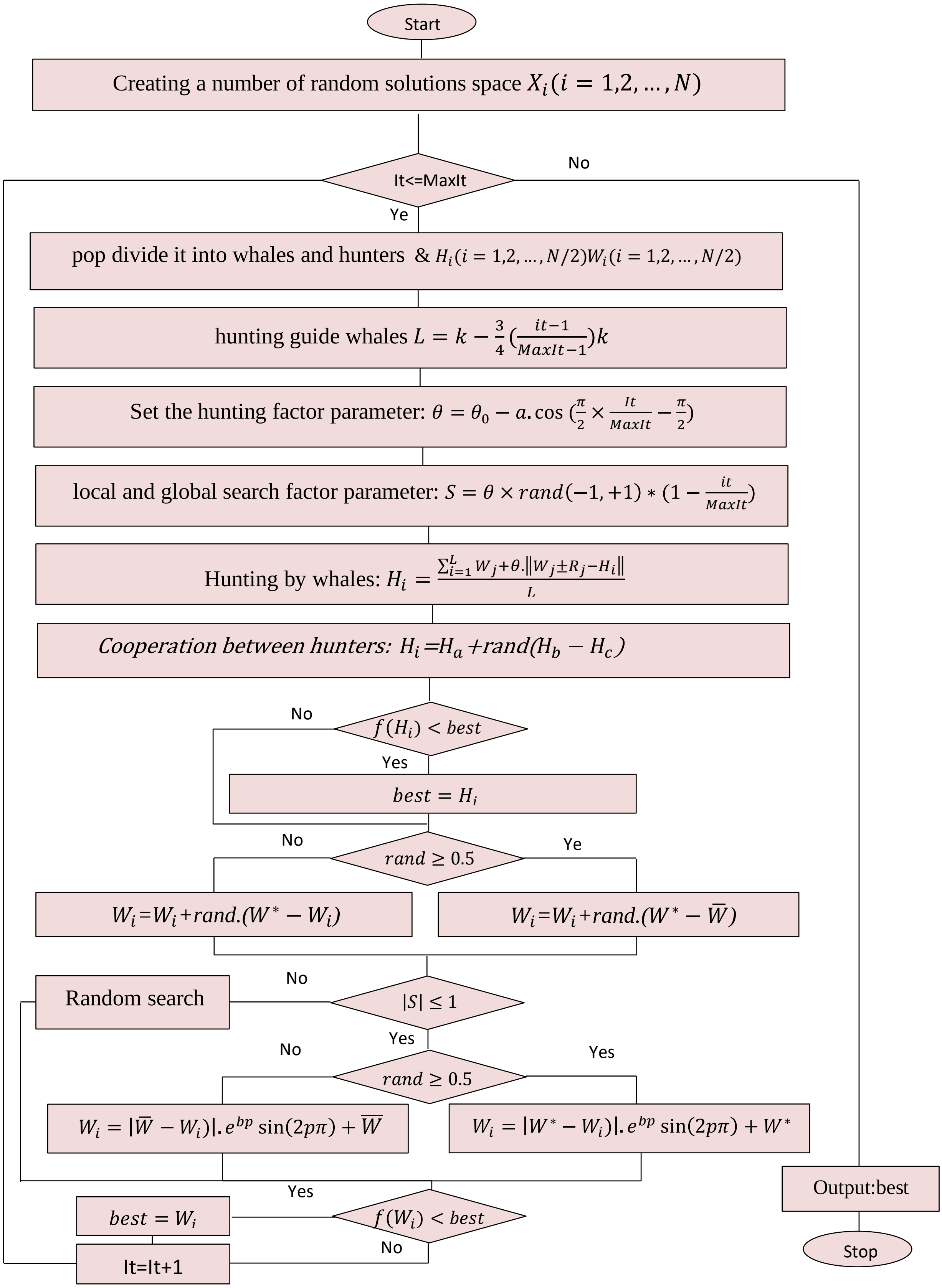

Figure 7: The Human-Whale Cooperation Optimization (HWO) Algorithm

3 Implementation results

To implement the human-whale cooperation optimization algorithm and other metaheuristic algorithms, MATLAB 2013 programming software or higher versions of this software and Windows 7 operating system with an Intel 5-core processor and 4 GB of memory are used. In order to measure the efficiency of the proposed method, a set of evaluation functions is used in MATLAB software. For this purpose, the proposed algorithm, along with other metaheuristic algorithms on each of the benchmark functions, performs the desired algorithms as many times as the experiment is repeated and the average convergence diagram is calculated. In order to provide the possibility of comparison on each benchmark function of the evaluation chart, the mean convergence of all metaheuristic algorithms used is analyzed graphically. It is also used to evaluate the average proposed algorithm (AVE), standard deviation (STD), and Wilcoxon test rank. In experiments, the proposed method in these indicators should be compared and evaluated with other metaheuristic algorithms. The proposed method in this paper has been compared with the proposed metaheuristic algorithm and especially swarm intelligence such as the grey wolf optimizer algorithm, salp swarm algorithm, butterfly optimization algorithm, the grasshopper optimization algorithm, and whale optimization algorithm in statistical indicators of mean error, rank, and standard deviation. In the experiments, an evaluation function is selected and each of these metaheuristic algorithms is performed 50 times on the evaluation function with a population equal to 30, the number of repetitions of 100 or 200. In the next step, the mean error, rank, and standard deviation for each of them are calculated and finally, the experiments are analyzed on a number of benchmark functions. In the experiments, the implementation parameters of the proposed method such as search radius, the base parameter for rotational motion and delta are considered equal to $R = 2$, $b = 1$ and $\theta_0 = 3$, respectively. In the particle swarm optimization algorithm, the inertial coefficient is 0.8, and the individual and swarm learning coefficients are both 2. In the differential evolution algorithm, the probability of combining is 0.2 and the minimum and maximum beta coefficients are 0 and 2, respectively. In the grasshopper optimization algorithm, the maximum and minimum values of the C coefficient are 1 and 0.00001, respectively, and on the other hand, the coefficients f and l in this algorithm are 0.5 and 1.5, respectively. The attractiveness coefficients and power in the butterfly optimization algorithm, which are displayed with c and a, are set at 0.1 and 0.01, respectively. The parameters used in the whale optimization algorithm such as b and l are also considered to be two random numbers in the range $[0, 1]$.

3.1 Evaluation functions

To evaluate meta-heuristic algorithms, a set of standard benchmark functions is usually required to evaluate the meta-heuristic algorithm. A number of benchmark functions used in this study in Table 1, Table 2 and Table 3 are shown with their optimal criteria and range.

Table 1: Unimodal functions

Function	Dimension	Range	$f_{\min}$
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30,100, 500, 1000	$[100, 100]$	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30,100, 500, 1000	$[10, 10]$	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	30,100, 500, 1000	$[100, 100]$	0
$f_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30,100, 500, 1000	$[100, 100]$	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	30,100, 500, 1000	$[30, 30]$	0
$f_6(x) = \sum_{i=1}^{n} \left([x_i + 0.5] \right)^2$	30,100, 500, 1000	$[100, 100]$	0
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{rand}\,[0, 1)$	30,100, 500, 1000	$[128, 128]$	0

Benchmark functions have been used as standard target functions to measure the efficiency of a meta-heuristic algorithm in many studies, so in this study, these criteria are used to evaluate the convergence of the algorithms. Benchmark functions are a set of mathematical functions that are essentially considered a function of cost and finding the national minimum in them is the main goal. Figure 8 shows one example of single-objective functions in three-dimensional mode and shows one example of multi-objective evaluation functions in MATLAB programming environment [23]:

3.2 HWO analysis

Analysis of problem space and benchmark functions show that evaluation functions such as Ackley, unlike e functions such as Sphere, have a more complex problem space, and this makes meta-algorithms more challenging. The main reason for the complex space of these benchmark functions can be considered in the presence of multiple local

Table 2: Multi-modal functions

Function	Dimension	Range	$f_{\min}$
$f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30,100, 500, 1000	$[500, 500]$	$-418.9829\times n$
$f_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right]$	30,100, 500, 1000	$[5.12, 5.12]$	0
$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right) + 20 + e$	30,100, 500, 1000	$[32, 32]$	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30,100, 500, 1000	$[600, 600]$	0
$f_{12}(x) = \frac{\pi}{n}\left\{10\sin\left(\pi y_1\right) + \sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right] + (y_n-1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	30,100, 500, 1000	$[50, 50]$	0
$y_i = 1 + \frac{x_i+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0-a & < x_i < a \\ k(-x_i-a)^m & x_i < -a \end{cases}$			
$f_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right] + (x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30,100, 500, 1000	$[50, 50]$	0

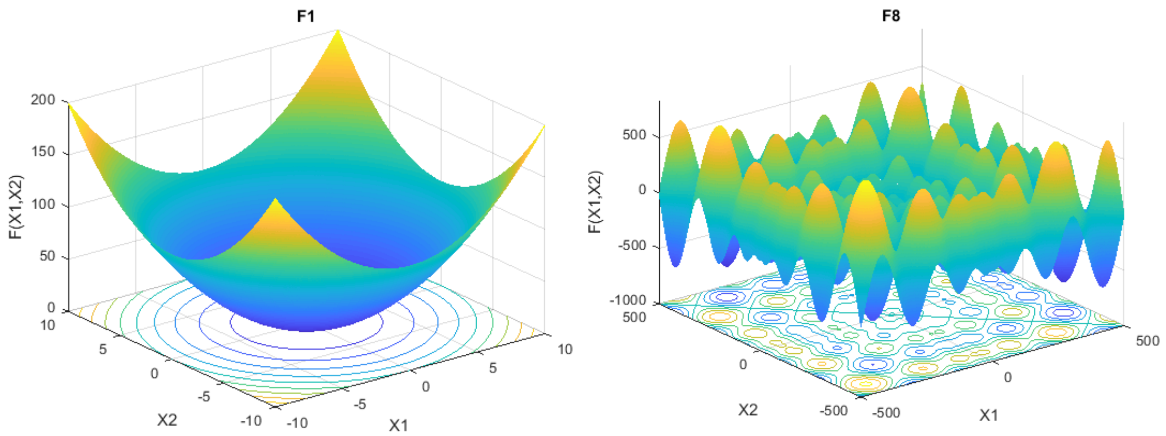

Figure 8: Three-dimensional mode displays two benchmark functions, F1 and F10

optimal that is located around the global optimization and finding the global optimal in them is not easy and there is a possibility of their deception in the local optimal. Here are two examples of implementing the proposed algorithm on the Ackley function, and the population size is 30 and the number of iterations is 200, and each experiment is repeated 50 times, and in each experiment, the different values of the parameter an in Figure 4 diagrams are used to show the effect of this parameter on the convergence of the proposed algorithm. In the diagram on the Figure 9, the value of the parameter $a = 0.5$, $a = 1$, $a = 1.5$ and $a = 2$ with a high slope. The analysis of the two figures shows that the proposed method is constantly reducing the error of global calculation based on iteration, and this reduction of this error also depends on the parameter a, and it is observed that for $a = 1.5$ the error of the proposed method is less and used. The low slope has a greater effect on the accuracy of the proposed method in reducing a by iteration.

Comparison and analysis of global optimal calculation error in different functions with metaheuristic algorithms with different approaches have also been compared in Figure 10, and the error has been compared in terms of iteration

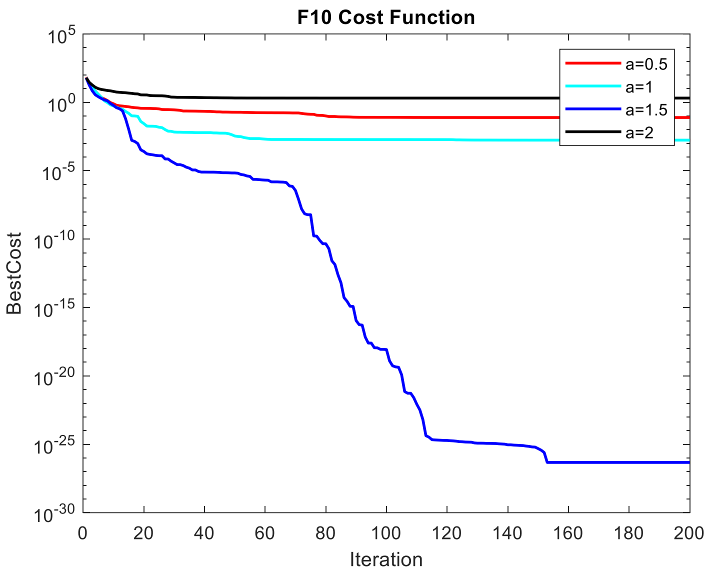

Figure 9: The effect of a high-slope parameter on reducing the optimal calculation error

in them. According to the experiments performed in the convergence diagrams of the proposed method and other metaheuristic algorithms, it can be concluded that in most experiments, the error of calculating the global optimal by the proposed algorithm compared to the butterfly optimization algorithm, grasshopper optimization, whale optimization, particle swarm optimization, and firefly algorithms are less. In most cases, the slope of the reduction in the calculation error of the global optimal in different benchmark functions in the proposed method is higher. In other words, the slope of the reduction in the error of the human-whale cooperation optimization algorithm to achieve the optimal solutions is higher than metaheuristic algorithms. In more complex benchmark functions, metaheuristics and proposed algorithms are more challenging to find the optimal error, but in general, the proposed method is less involved in local optimization because the slope of the error reduction is higher than other algorithms. According to the diagrams, the reduction of the optimal calculation error in terms of iteration of meta-heuristic algorithms can provide the following results:

- The proposed algorithm reduces the optimal calculation error in terms of iteration. This error reduction is reduced by a steeper slope than other methods. The rapid reduction of error in terms of iteration indicates that the local and global search mechanism of the proposed algorithm is efficient.
- The reason for the success of the proposed method in reducing the error rate is that the proposed algorithm leads the population to global optimizations with high acceleration.
- The proposed algorithm in functions such as F10 that have local optimization has been able to reduce the error reduction slope and this indicates that the algorithm is not caught in local optimization or its convergence rate to local optimization is low.
- Our experiments show that the convergence rate of the proposed algorithm on the evaluation functions with local optimization is about 4.36%, and this rate is 7.64% in the gray wolf optimization algorithm and 8.67% in the butterfly optimization algorithm.

3.3 Qualitative analysis

The qualitative analysis of the human-whale cooperation optimization algorithm uses four evaluation functions F1, F9, F10, and F11. For qualitative analysis, we can analyze the changes in the history of solutions to the optimal solution and also analyze and evaluate the average competency of population members as well as the convergence diagram in terms of iteration. Figure 11 shows the analysis of the F1 and F10 functions with a population size of 30 and an iteration number of 20. The diagrams of the history of whales and hunters' movement towards the optimal solution in the evaluation functions used show that the human-whale cooperation optimization algorithm has a clear

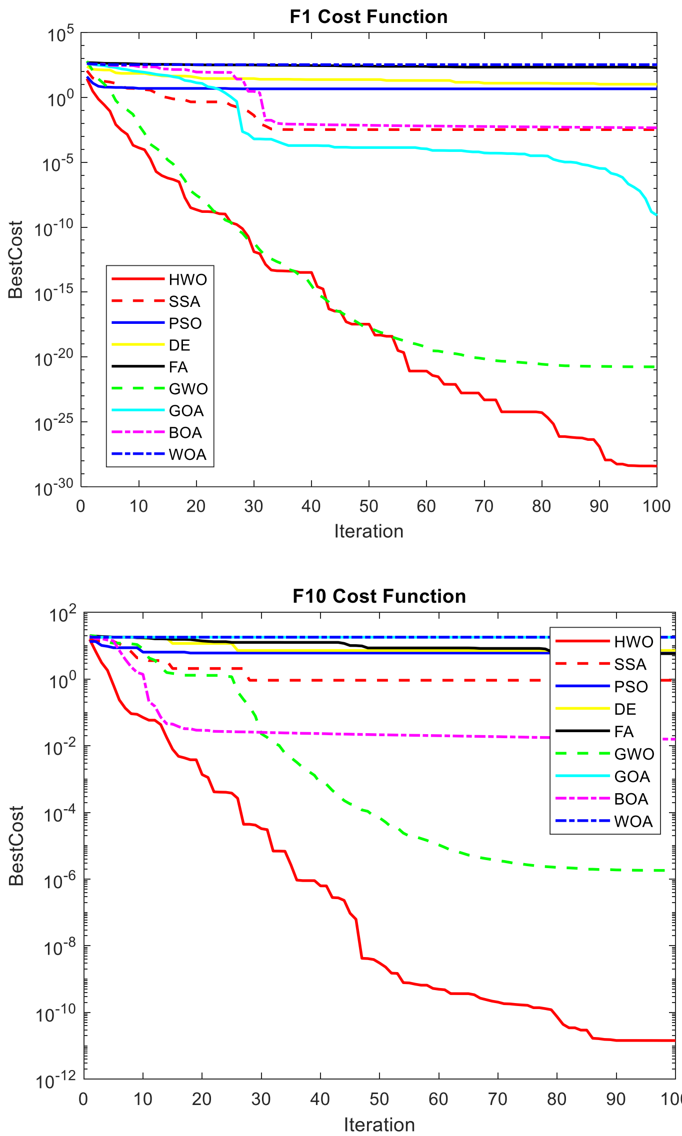

Figure 10: The output of the proposed algorithm and other algorithms on several functions

and similar pattern for guiding the population to the optimal solution. The proposed algorithm first tries to search the space around the optimal solution to find the most optimal solution, and then, by repeating the algorithm, the members of the population are sent to the optimal solution, which here are the coordinates of the origin. The trajectory diagram helps us to better understand the behavior of whales and predators to find the optimal solution. The path diagram or trajectory diagram shows how well the best solution is trying to reach and change the optimal solution, and this change of direction is ultimately such that the optical member is directed to the optimal solution.

3.4 Sustainability

One of the important factors that show how accurate and practical a meta-heuristic algorithm is is the sustainability of that algorithm. The sustainability of a meta-heuristic algorithm can be analyzed using indicators such as standard deviation(STD), which are analyzed in the next section of these calculations. Another important factor in analyzing the sustainability of metaheuristic algorithms is the error of calculating the global optimal in the metaheuristic algorithm by increasing the dimensions of the problem and if it does not decrease much, it indicates the intelligence of the algorithm. In other words, by increasing the dimensions of the problem, if a meta-heuristic algorithm increases its computational error, it will not be highly stable. Here, to evaluate the sustainability of the human-whale cooperation optimization algorithm, the behavior of the algorithm and the error of calculating the global optimal can be examined by increasing the dimensions, and therefore in this section, the dimensions are changed from 2 to 100 other metaheuristic algorithms are compared with a population of 30 and a repeat of 100. Analyzing the diagrams in Figure 12 on two functions shows that the error of calculating the proposed method and other algorithms increases with increasing dimensions, but this increase in the proposed method is less than other algorithms and this shows the proposed algorithm compared to the algorithm. This suggests that the proposed algorithm is more stable than competing algorithms because the error-increasing diagrams are below the other diagrams.

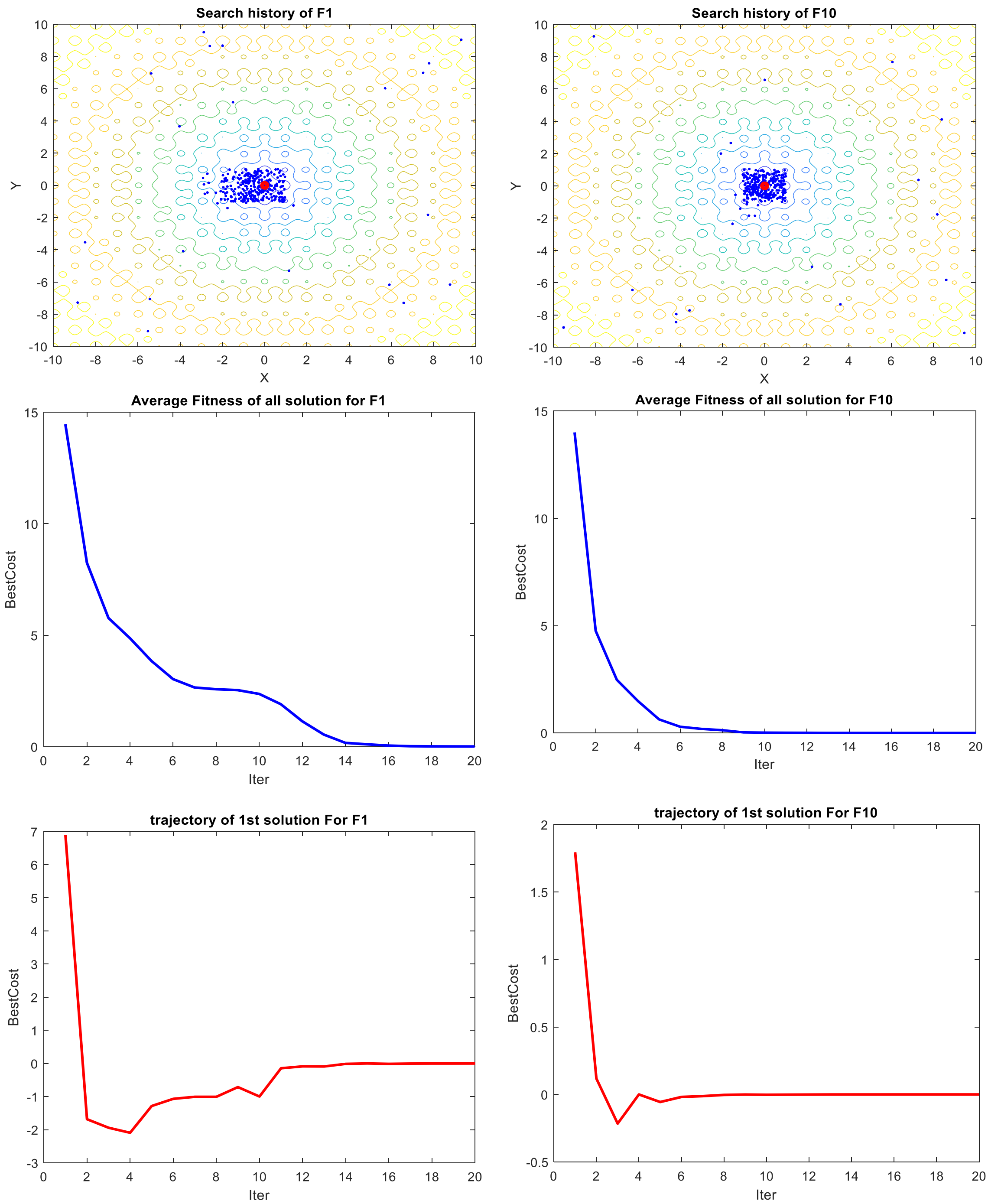

Figure 11: The earch history diagrams, trajectory diagrams, and the average merit chart of population members of the human-whale cooperation optimization algorithm in the F1 and F10 functions,

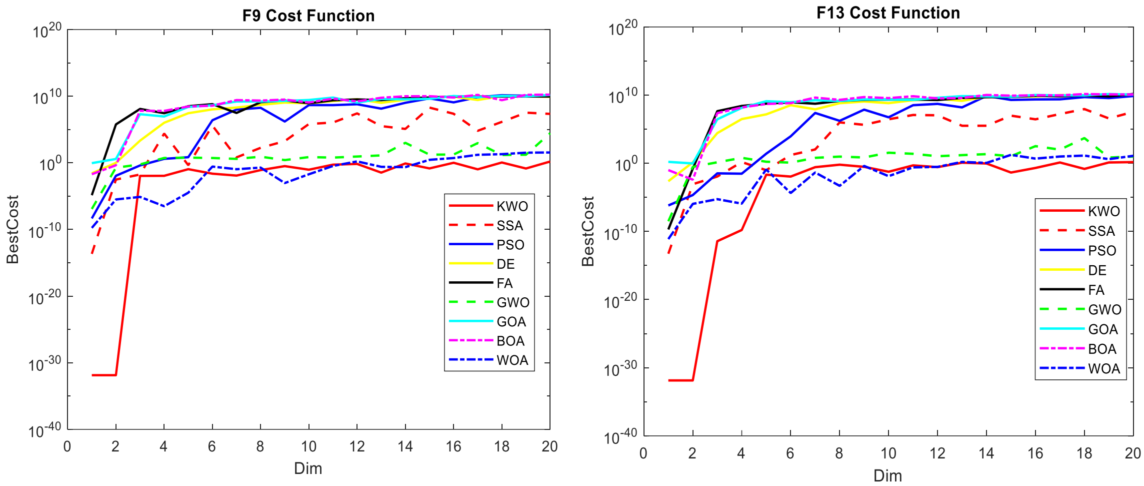

Figure 12: Sustainability of human-whale cooperation optimization algorithm in functions

3.5 Analysis

To evaluate the proposed method in this section, three criteria of mean error, rank, and deviation are measured and the results of experiments on F1 to F7 benchmark functions in Table 3 and the results of experiments on F8 to F13 benchmark functions in Table 4 and in Finally, the results of experiments on F14 to F23 benchmark functions are also listed in Table 5, so that the proposed algorithm can be compared with other metaheuristic algorithms in these three metrics. According to the experiments performed on the 23 evaluation functions, it can be concluded that the human-whale cooperation optimization algorithm performed 19 times better than other metaheuristic algorithms in the average error evaluation index, and in the ranking index, in most experiments, it ranks better than particle swarm optimization algorithm, evolutionary difference, firefly algorithm, gray wolf algorithm, slap swarm algorithm, butterfly optimization algorithm, grasshopper algorithm, and whale optimization algorithm. The closer the rank index is to the number one, the more accurate the metaheuristic algorithm is in finding the optimal solution, and if the algorithm rank in a benchmark function is 8, then the algorithm's performance is poor and has the worst accuracy in detecting the optimal solution. Analysis of the proposed algorithm and other algorithms shows that the gray wolf optimization algorithm after the proposed method in most cases has less error in finding the optimal solution and is in second place. Analysis of the STD index also shows that the standard deviation of the experimental results of the proposed method in finding the optimal solution is less than most other algorithms in most experiments, and this shows the greater stability of the algorithm to find the optimal solution. The stability of the proposed method is higher than other methods because, with increasing dimensions, there is no significant reduction in calculating the error of the proposed method and this indicates that the proposed method can be used for complex problems with high dimensions. The proposed method has a lower mean error for optimal calculation in 19 of the 23 benchmark functions, and this value is equal to 82.60% of the evaluation functions.

Table 3: Comparison of error in uni-modal functions

CostFunction	Metric	HWO	BOA	PSO	DE	FA	GWO	GOA	SSA
F1	AVG	**1.58E-12**	0.4211	1.7238	1.7777	2.2135	1.58E-09	0.007306	0.45696
	STD	**4.9E-12**	0.73235	1.6998	1.34357	1.68752	1.3E-11	0.006741	0.191704
	RANK	**1.3**	4.2	6	6.1	6.7	1.7	3	4
F2	AVG	4.46E-06	0.013497	39.96887	33.31704	30.25814	**2.31E-07**	36.16126	6.389916
	STD	9.95E-06	0.007186	41.14289	14.59301	3.122407	**4.64E-07**	5.781187	2.737418
	RANK	1.8	3	6.1	6.4	6.3	**1.2**	7.2	4
F3	AVG	**8.59E-10**	7434.709	7591.773	8753.185	10924.27	0.648974	14351.71	1743.068
	STD	**1.49E-09**	9169.176	3037.575	2693.226	2454.431	0.917841	2492.443 1157.252	
	RANK	1	4.3	5.2	5.6	6.2	2.5	7.6	3.6
F4	AVG	**1.41E-07**	0.012973	44.23104	56.92657	56.17796	0.002077	60.25541	17.21565
	STD	**2.79E-07**	0.011302	18.84296	3.554017	9.638731	0.002014	6.071298	5.171652
	RANK	1	2.9	5.8	6.4	6.9	2.1	6.8	4.1
F5	AVG	**8.470349**	41883365	2291061	8809313	25328510	8.908562	26912013	120593.4
	STD	**0.4323**	16126576	1689901	5875508	9620100	0.063418	16507134	234807.8
	RANK	**1.1**	7.4	4.1	5	6.6	1.9	6.8	3.1
F6	AVG	**7.07E-06**	10367.94	3525.608	10778.69	14340.91	0.624021	13141.29	742.9453
	STD	**1.35E-05**	9539.639	1569.957	2834.784	2188.068	0.251607	2925.132	1474.074
	RANK	**1**	5.6	4.3	5.8	7.1	2	6.7	3.5
F7	AVG	**0.001201**	0.005538	1.223793	3.288311	0.008758	0.002906	6.106226	0.129071
	STD	**0.000693**	0.005124	0.584899	1.463638	0.004522	0.004446	3.082125	0.102831
	RANK	1.4	2.9	6.1	7.3	3.6	2.1	7.6	5

Table 4: Comparison of error in multi-modal functions

CostFunction	Metric	HWO	BOA	PSO	DE	FA	GWO	GOA	SSA
F8	AVG	-39200.9	-2688.8	-5.1E+78	-3538.19	-4555.92	-1.1E+07	**-2436.81**	-134481
	STD	**23.8431**	430.8567	1.61E+79	372.9681	489.3166	17690867	429.9333	200986.9
	RANK	3.3	2.4	7.7	6	5	7.3	1	3.3
F9	AVG	**1.92E-10**	0.005778	81.7201	94.43346	80.17709	2.810926	91.44911	48.81308
	STD	**3.01E-10**	0.005764	16.40374	6.658141	14.71925	8.888929	15.75242	9.928804
	RANK	1	2.9	6.3	7.2	5.5	2.1	6.9	4.1
F10	AVG	**4.36E-06**	0.013166	15.73703	19.0244	19.24043	1.92E-06	18.60506	10.30364
	STD	**7.95E-06**	0.009876	1.818641	0.367317	0.390993	1.82E-06	0.118468	1.76693
	RANK	**1.4**	3	5	7.1	7.6	1.6	6.3	4
F11	AVG	**1.74E-12**	0.000977	33.93312	94.14153	124.1417	0.098187	117.5729	3.159438
	STD	**4.03E-12**	0.002389	13.07585	29.03922	33.73477	0.209564	36.02208	1.856087
	RANK	1	2.5	5.1	6.3	7.4	2.2	7.2	4
F12	AVG	**3.49E-05**	56558213	1912292	16082252	52310071	0.173418	70835550	2790.213
	STD	**4.38E-05**	27953267	1919813	9150143	36315375	0.140097	40341973	8124.223
	RANK	1	6.9	4	5.3	6.5	2	7.3	3
F13	AVG	0.35605	1.87E+08	4624338	61421341	98101300	**0.288271**	1.24E+08	73584.81
	STD	0.3558	91183793	4917286	22527323	41976536	**0.121274**	82618511	95755.5
	RANK	1.5	7.7	3.9	5.4	6.4	1.5	6.5	3.1

3.6 HWO for classical engineering problems

The pressure vessel design problem for gas storage capsules and oil metals that are used in refineries as distillation towers is very important in the oil and gas industry. Figure **??** shows this problem, which has four continuous variables. In this case, the goal of the designers is to increase the bearing capacity of the pressure tank as much as possible and at the same time minimize the cost of designing the tank. Minimizing the cost function of the Pressure vessel design problem allows a solid structure to be provided at the lowest possible cost and high strength. The objective function defines the problem of optimizing the gas pressure tank according to the relationship and according to the variables of body radius, body length, body thickness, and end thickness of the cap, and this issue has a number of conditions and limitations that are stated following the objective function. where z_1, z_2, z_3 and z_{14} are the thickness of the body, the thickness of the cap, the radius of the cylinder, and the length of the cylinder, respectively (Eq. 3.24).

$$f = [z_1 z_2 z_3 z_4] = [T_s T_h R L], \tag{3.24}$$

Table 5: Comparison of errors of the proposed algorithm and other algorithms with multi-objective functions

CostFunction	Metric	HWO	BOA	PSO	DE	FA	GWO	GOA	SSA
F14	AVG	**1.191088**	9.728288	1.273478	3.182885	1.636128	3.512561	10.80063	1.735439
	STD	1.559269	3.712016	**0.43363**	1.565201	0.978683	2.913483	5.62999	1.129537
	RANK	1.55	7.4	3	5.2	3.1	4.9	6.95	3.9
F15	AVG	**0.003216**	0.019839	0.001593	0.009264	0.001263	0.005069	0.057421	0.002572
	STD	**0.001351**	0.025874	0.000506	0.004326	0.000636	0.005047	0.055058	0.002382
	RANK	2.9	6.2	3.4	6.1	2.3	4.5	7.7	2.9
F16	AVG	**-1.03163**	-0.82869	-1.03151	-1.00171	-1.03165	**-1.03163**	-0.78678	-1.03019
	STD	**2.05E-10**	0.345936	0.000216	0.015159	1.48E-05	3.41E-07	0.394245	0.002681
	RANK	1.2	6.7	4.8	7.3	3.8	3.2	3.6	5.4
F17	AVG	**0.397987**	0.459846	0.39846	0.433108	0.397899	0.407625	0.466542	0.404427
	STD	**7.86E-15**	0.10561	0.000804	0.030689	7.44E-06	0.015732	0.133871	0.00882
	RANK	1	6.4	4.1	6.8	3	5.2	4.1	5.4
F18	AVG	**3.000001**	31.78297	3.000573	3.664853	3.000213	9.151628	16.5	3.001485
	STD	**1.65E-06**	33.98329	0.000643	0.840048	0.000207	11.36556	26.23928	0.003043
	RANK	1.3	7.6	3.8	6.5	3.5	6.1	3.4	3.8
F19	AVG	**-3.86217**	-3.6673	-3.86233	-3.81942	-3.86268	-3.75533	-3.65006	-3.82323
	STD	**4.92E-05**	0.149103	0.000522	0.02507	6.61E-05	0.10149	0.23407	0.111563
	RANK	1.5	6.7	3	5.7	2.4	6.4	6.7	3.6
F20	AVG	**-3.25586**	-2.71567	-3.16172	-2.78922	-3.23362	-2.82325	-1.70059	-3.19307
	STD	0.076626	0.199936	0.06522	0.189005	**0.052918**	0.21271	0.659644	0.151489
	RANK	2.1	6.4	3.2	5.9	2.2	5.6	7.8	2.8
F21	AVG	-8.87602	-3.23494	-3.79125	-1.3858	**-8.93828**	-4.37105	-0.88891	-8.11637
	STD	2.648484	2.184127	1.939383	**0.513734**	2.094238	1.53979	0.688367	2.81246
	RANK	2.5	5.4	4.6	7	2	4.4	7.5	2.6
F22	AVG	**-8.74509**	-2.6207	-4.54106	-1.851	-7.34531	-5.13085	-1.27165	-8.10113
	STD	**0.274113**	1.319504	2.66781	0.673575	3.049751	2.124433	0.866043	2.685062
	RANK	2.5	5.8	4.4	7	2.7	3.5	7.5	2.6
F23	AVG	-8.24377	-2.55884	-5.58245	-2.09469	-8.84752	-4.61002	-1.62696	-6.81999
	STD	2.982962	2.251974	3.205711	0.5133	2.747315	2.330573	1.286726	3.682492
	RANK	2.1	6.3	4.3	6.3	2.3	3.9	7.4	3.4

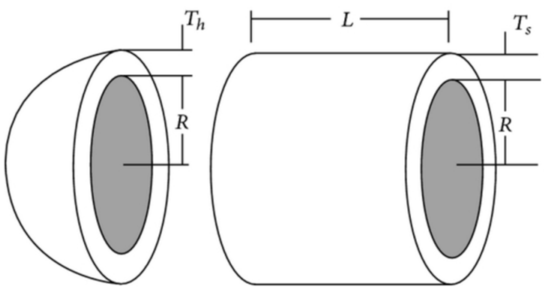

Figure 13: Pressure vessel design problem

$$\begin{aligned} \min \quad & (\overline{z}) = 0.6224z_1z_3z_4 + 1.7781z_2z_3^2 + 3.1661z_1^2z_4 + 19.84z_1^2z_3 \\ \text{Subject to :} \quad & g_1\left(\overrightarrow{z}\right) = -z_1 + 0.0193z_3 \leq 0 \\ & g_2\left(\overrightarrow{z}\right) = -z_3 + 0.00954z_3 \leq 0 \\ & g_3\left(\overrightarrow{z}\right) = -\Pi z_3^2z_4 - \tfrac{4}{3}\Pi z_3^3 + 1,296,000 \leq 0 \\ & g_4(\overrightarrow{z}) = z_4 - 240 \leq 0, \end{aligned}$$

In Table 6, the solution to the pressure vessel design problem and finding its optimal parameters as well as the value of the cost function are compared by the proposed method with other methods and it is observed that the proposed method calculates more optimal values. The optimal design of spring as an important component in power transmission systems will transfer the maximum possible power with minimal design costs (Figure 14):

The cost function of this problem is defined and modeled as the following relation and at the same time its conditions and limitations are formulated by Eq. (3.25):

$$f = [z_1z_2z_3] = [dDN], \tag{3.25}$$

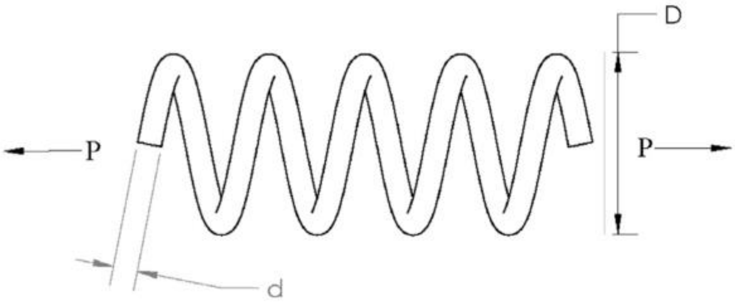

Figure 14: Tension/compression spring

$$\begin{aligned}
\text{Min} \quad & f\left(\overrightarrow{z}\right) = \left(z_3+2\right) z_2 z_1^2, \\
\text{Subjectto}: \quad & g_1\left(\overrightarrow{z}\right) = 1 - \frac{z_2^3 z_3}{71785 z_1^4} \le 0 \\
& g_2\left(\overrightarrow{z}\right) = \frac{4z_2^2 - z_1 z_2}{12566\left(z_2 z_1^3 - z_1^4\right)} + \frac{1}{5108 z_1^2} \le 0 \\
& g_3\left(\overrightarrow{z}\right) = 1 - \frac{140.45 z_1}{z_2^2 z_3} \le 0 \\
& g_4\left(\overrightarrow{z}\right) = \frac{z_1+z_2}{1.5} - 1 \le 0
\end{aligned}$$

z_1, z_2 and z_3 are the diameter of each spring ring, the diameter of the spring and the number of spring rings per meter, respectively. Solving the Tension/compression spring problem with the HWO algorithm and comparing its optimal value with other meta-heuristic methods according to Table 7, shows that the HWO algorithm provides more optimal values than other methods. Three-bar truss design with the help of three rods is also one of the practical optimization problems that are used in the construction industry and the purpose of this problem is according to Figure 15, the purpose of designing this structure with the least possible weight and high resistance:

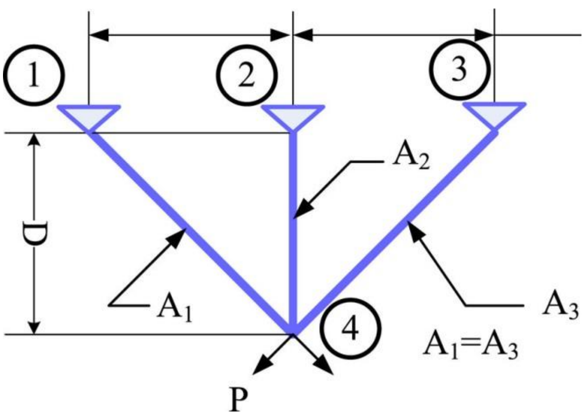

Figure 15: Three-bar truss problem

The cost function of Three-bar truss problem and the variables used in it can be displayed as follows, and on the other hand, this problem has a number of limitations and conditions that are implemented on the solutions and the cost function (Eq. (3.26)):

$$f = [x_1 x_2] = [A_1 A_2], \tag{3.26}$$

$$\begin{aligned}
\text{Minimize} \quad & f\left(\overrightarrow{X}\right) = \left(2\sqrt{2}X_1 + X_2\right) \times L, \\
\text{Subject to} \quad & g_1\left(\overrightarrow{X}\right) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \le 0, \\
& g_2\left(\overrightarrow{X}\right) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \le 0, \\
& g_3\left(\overrightarrow{X}\right) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \le 0, \\
\text{Variable range:} \quad & 0 \le x_1, x_2 \le 1, \\
\text{where:} \quad & L = 100cm, P = 2KN/cm^2, \sigma = 2KN/cm^2
\end{aligned}$$

The goal of this problem is to find solutions that minimize the function of the goal. In this case, the solution is the length of the bars, which need to be optimally selected. Experiments and the implementation of the proposed method on this problem show that the proposed method has been able to optimize or minimize the cost function compared to other metaheuristic algorithms according to Table 8.

Table 6: Comparison of results for Pressure vessel design problem[15]

Algorithms	Ts (x1)	Th (x2)	R(x3)	L(x4)	Optimal
HWO	**0.810245**	**0.400352**	**41.7845**	**178.0012**	**5924.2638**
HHO	0.81758383	0.4072927	42.09174576	176.7196352	6000.46259
GWO	0.8125	0.4345	42.089181	176.758731	6051.5639
GA	0.812500	0.437500	42.097398	176.654050	6059.9463
HPSO	0.812500	0.437500	42.0984	176.6366	6059.7143
G-QPSO	0.812500	0.437500	42.0984	176.6372	6059.7208
WEO	0.812500	0.437500	42.098444	176.636622	6059.71
IACO	0.812500	0.437500	42.098353	176.637751	6059.7258
BA	0.812500	0.437500	42.098445	176.636595	6059.7143
MFO	0.8125	0.4375	42.098445	176.636596	6059.7143
CSS	0.812500	0.437500	42.103624	176.572656	6059.0888
ESs	0.812500	0.437500	42.098087	176.640518	6059.7456
CPSO	0.812500	0.437500	42.091266	176.746500	6061.0777
BIANCA	0.812500	0.437500	42.096800	176.6580 0 0	6059.9384
MDDE	0.812500	0.437500	42.098446	176.636047	6059.701660
DELC	0.812500	0.437500	42.0984456	176.6365958	6059.7143
WOA	0 .812500	0 .437500	42 .0982699	176 .638998	6059 .7410
GA3	0.812500	0.437500	42.0974	176.6540	6059.9463

Table 7: Comparison of results for tension/compression spring problem [15]

Algorithms	d	*D*	*N*	Optimal
HWO	**0.051073**	**0.342851**	**11.2542**	**0.0118533**
HHO	0.051796393	0.359305355	11.138859	0.0126654
SSA	0.051207	0.345215	12.004032	0.0126763
TEO	0.051775	0.3587919	11.16839	0.012665
MFO	0.051994457	0.36410932	10.868422	0.0126669
SFS	0.051689061	0.356717736	11.288966	0.01266523
GWO	0.05169	0.356737	11.28885	0.012666
WOA	0 .051207	0 .345215	12 .004032	0 .0126763
GA2	0.051480	0.351661	11.632201	0.012704
GA3	0.051989	0.363965	10.890522	0.012681
CPSO	0.051728	0.357644	11.244543	0.012674
DEDS	0.051689	0.356717	11.288965	0.012665
GSA	0.050276	0.323680	13.525410	0.012702

4 Conclusion

Metaheuristic algorithms inspired by the laws of nature and their mechanisms are a good and ideal way to solve optimization problems. Meta-heuristic algorithms and swarm intelligence algorithms, unlike conventional methods of solving optimization problems such as mathematical methods, do not require an information gradient, therefore they can be used to solve various and complex optimization problems. Meta-heuristic algorithms are classified into different categories such as evolutionary algorithms and swarm intelligence algorithms. Swarm Intelligence Algorithms have used group behavior of animals to solve the optimization problem. In this paper, the behavior of whales and their involvement with humans in fishing are used to model the HWO algorithm. Experiments on many benchmark functions and some optimization problems show the proposed method at least of particle swarm optimization algorithms, evolutionary difference, firefly algorithm, gray wolf algorithm, salp optimization algorithm, butterfly optimization algorithm, grasshopper optimization algorithm, and whale optimization algorithm more accurate in finding the optimal

Table 8: Comparison of results for Three-bar truss problem [15]

Algorithms	**x1**	**x2**	**Optimal**
HWO	**0.7887354**	**0.407078**	**263.7958599**
HHO	0.788662816	0.408283133832900	263.8958434
DEDS	0.78867513	0.40824828	263.8958434
MVO	0.78860276	0.408453070000000	263.8958499
GOA	0.788897555578973	0.407619570115153	263.895881496069
MFO	0.788244771	0.409466905784741	263.8959797
PSO-DE	0.7886751	0.4082482	263.8958433
SSA	0.788665414	0.408275784444547	263.8958434
MBA	0.7885650	0.4085597	263.8958522
CS	0.788	0.408	263.68

solution. In most experiments, the proposed method ranks first in finding the optimal solution with the least error. Due to the importance of optimization methods in increasing the accuracy of machine learning techniques, future research uses the HWO algorithm to select features and applications related to machine learning and detect fake pages on the Internet.

References

[1] H. Abedinpourshotorban, S.M. Shamsuddin, Z. Beheshti and D.N. Jawawi, *Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm*, Swarm Evolut. Comput. **26** (2016), 8–22.

[2] A.A. Asif, *Evolution of ant colony optimization algorithm: A brief literature review*, arXiv preprint arXiv:1908.08007, 2019.

[3] M.R. Bonyadi, *A theoretical guideline for designing an effective adaptive particle swarm*, IEEE Trans. Evolut. Comput. **24** (2019), no. 1, 57–68.

[4] M. Bordbar, A. Neshat, S. Javadi, B. Pradhan and H. Aghamohammadi, *Meta-heuristic algorithms in optimizing GALDIT framework: a comparative study for coastal aquifer vulnerability assessment*, J. Hydrology **585** (2020), 124768.

[5] M.C. Catalbas and A. Gulten, *Circular structures of puffer fish: A new metaheuristic optimization algorithm*, Third Int. Conf. Electric. Biomed. Engin. Clean Energy Green Comput. (EBECEGC), 2018, pp. 1–5.

[6] C. Chen, *RWFOA: a random walk-based fruit fly optimization algorithm*, Soft Comput. **24** (2020), no. 16, 12681–12690.

[7] S. Chowdhury, M. Marufuzzaman, H. Tunc, L. Bian and W. Bullington, *A modified ant colony optimization algorithm to solve a dynamic traveling salesman problem: a case study with drones for wildlife surveillance*, J. Comput. Design Engin. **6** (2019), no. 3, 368–386.

[8] E. Cuevas, F. Fausto and A. Gonzjlez, *The swarm method of the social-spider*, New Advancements in Swarm Algorithms: Operators and Applications. Springer, Cham, 2020. 111-137.

[9] G. Dhiman and V. Kumar, *Emperor penguin optimizer: A bio-inspired algorithm for engineering problems*, Knowledge-Based Syst. **159** (2018), 20–50.

[10] G. Dhiman and V. Kumar, *Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications*, Adv. Engin. Software **114** (2017), 48–70.

[11] G. Dhiman and V. Kumar, *Spotted hyena optimizer for solving complex and non-linear constrained engineering problems*, Harmony search and nature inspired optimization algorithms. Springer, Singapore, 2019. 857-867.

[12] E. Fadakar and M. Ebrahimi, *A new metaheuristic football game inspired algorithm*, 1st Conf. Swarm Intell. Evolut. Comput. (CSIEC), 2016, pp. 6–11.

[13] A. Faramarzi, M. Heidarinejad, S. Mirjalili and A.H. Gandomi, *Marine predators algorithm: A nature-inspired metaheuristic*, Expert Syst. Appl. **152** (2020), 113377.

[14] V. Hayyolalam and A.A.P. Kazem, *Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems*, Engin. Appl. Artific. Intell. **87** (2020), 103–249.

[15] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja and H. Chen, *Harris hawks optimization: Algorithm and applications*, Future Gen. Comput. Syst. **97** (2019), 849–872.

[16] A.A. Heidari, H. Faris, S. Mirjalili, I. Aljarah and M. Mafarja, *Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks*, Nature-Inspired Optim. (2020), 23–46.

[17] M.M. Islam, H. Shareef, A. Mohamed and A. Wahyudie, *A binary variant of lightning search algorithm: BLSA*, Soft Comput. **21** (2017), no. 11, 2971–2990.

[18] C.B. Kalayci, O. Polat and M.A. Akbay, *An efficient hybrid metaheuristic algorithm for cardinality constrained portfolio optimization*, Swarm Evolution. Comput. **54** (2020), 100–662.

[19] A. Kaveh and T. Bakhshpoori, *Water evaporation optimization: a novel physically inspired optimization algorithm*, Comput. Structures **167** (2016), 69–85.

[20] T. Khurshaid, A. Wadood, S.G. Farkoush, C.H. Kim, J. Yu and S.B. Rhee, *Improved firefly algorithm for the optimal coordination of directional overcurrent relays*, IEEE Access **7** (2019), 78503–78514.

[21] Z. Lei, S. Gao, S. Gupta, J. Cheng and G. Yang, *An aggregative learning gravitational search algorithm with self-adaptive gravitational constants*, Expert Syst. Appl. **152** (2020), 113396.

[22] Q. Liu, J. Li, L. Wu, F. Wang and W. Xiao, *A novel bat algorithm with double mutation operators and its application to low-velocity impact localization problem*, Engin. Appl. Artific. Intell. **90** (2020), 103–505.

[23] S. Mirjalili, *Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems*, Neural Comput. Appl. **27** (2016), no. 4, 1053-1073.

[24] S. Mirjalili, I. Aljarah, M. Mafarja, A.A. Heidari and H. Faris, *Grey Wolf optimizer: theory, literature review, and application in computational fluid dynamics problems*, Nature-Inspired Optim. (2020), 87–105.

[25] S. Mirjalili and A. Lewis, *The whale optimization algorithm*, Adv. Engin. Software **95** (2016), 51–67.

[26] S. Mirjalili, S. M. Mirjalili, S. Saremi and S. Mirjalili, *Whale optimization algorithm: theory, literature review, and application in designing photonic crystal filters*, Nature-Inspired Optim. (2020), 219-238.

[27] S.H.S. Moosavi and V.K. Bardsiri, *Poor and rich optimization algorithm: A new human-based and multi populations algorithm*, Engin. Appl. Artific. Intell. **86** (2019), 165–181.

[28] A. Mukherjee and D. De, *Octopus algorithm for wireless personal communications*, Wireless Person. Commun. **101** (2018), no. 1, 531-565.

[29] M.G. Omran and S. Al-Sharhan, *Improved continuous Ant Colony Optimization algorithms for real-world engineering optimization problems*, Engin. Appl. Artific. Intell. **85** (2019), 818–829.

[30] M. Orujpour, M. R. Feizi-Derakhshi and T. Rahkar-Farshi, *Multi-modal forest optimization algorithm*, Neural Comput. Appl. **32** (2020), no. 10, 6159–6173.

[31] C.M. Rahman and T. A. Rashid, *A survey on dragonfly algorithm and its applications in engineering*, arXiv preprint arXiv: (2020), 12-126.

[32] M.A. Shaheen, H.M. Hasanien, S.F. Mekhamer and H.E. Talaat, *Optimal power flow of power systems including distributed generation units using sunflower optimization algorithm*, IEEE Access **7** (2019), 109289–109300.

[33] A.K. Shukla, P. Singh and M. Vardhan, *An adaptive inertia weight teaching-learning-based optimization algorithm and its applications*, Appl. Math. Modell. **77** (2020), 309–326.

[34] H. Whitehead and R. Reeves, *Killer whales and whaling: the scavenging hypothesis*, Bio. Lett. **1** (2005), no. 4, 415–418.

[35] C. Yang, J. Ji and S. Li, *Stability analysis of chemotaxis dynamics in bacterial foraging optimization over multi-dimensional objective function*, Soft Comput. **24** (2020), no. 5, 3711–3725.